

REMARKS

This amendment responds to the office action mailed July 29, 2004. In the office action the Examiner:

- rejected claims 1, 2, 5-7, 11, 12, and 15-17 under 35 U.S.C. 102(a) as anticipated by Burger et al. (US 5,097,533);
- rejected claims 3, 8-10, 13, 18-20, and 21 under 35 U.S.C. 103(a) as being unpatentable over Burger et al (US 5,097,533) and Dobbs (US 6,039,426);
- rejected claims 4 and 14 under 35 U.S.C. 103(a) as being unpatentable over Burger et al. (US 5,097,533), Dobbs (US 6,039,426) and Ball (US 6,189,047);
- rejected claim 22 under 35 U.S.C. 103(a) as being unpatentable over Burger et al. (US 5,097,533), Dobbs (US 6,039,426) and Pham et al. (US 5,524,253)

After entry of this amendment, the pending claims are: claims 1-22.

Claim rejections under 35 U.S.C. 102(a)

Claim 1, as amended, is directed to a method of automatically learning control sequences associated with a plurality of computer application programs. The method includes extracting first and second sets of control sequences from first and second computer application programs, respectively, loading the first and second sets of control sequences into a data structure, and executing the first and second computer application programs using the first and second sets of control sequences stored in the data structure.

Figs. 12 – 15 of the present application, in connection with specification from page 31 to page 38, illustrate one embodiment of claim 1, a User-Assisted LearnApp application. The LearnApp has a GUI 1206 (Fig. 12B) for receiving instructions from a user on how to execute user-specified computer application programs in a user-specified manner. For example, if the user clicks on the “Learn All” button 1220, the LearnApp executes a list of computer application programs that have been selected by the user. If the user clicks on the “Learn Highlighted” button 1222, the learnApp executes the computer application programs that have been highlighted by the user.

The LearnApp executes a computer application program in order to collect a set of control sequences from the computer application program, such as open/print/close document variables, and then applies them to the computer application program in a predefined manner

(see, e.g., steps 1320 – 1360 in Fig. 13). If the LearnApp is unable to determine some items of the control sequences, it prompts the user to provide required information by opening a dialog box (Fig. 15).

In contrast, Burger teaches a method for providing an interface between a plurality of computer application programs written in different languages and a system software such as a database manager. Fig. 3 of Burger discloses that multiple instances (60, 62) of a generic API interface a set of application programs with a system software (e.g., a database application) such that different application programs written in different languages are able to submit instructions to the software system, e.g., instruction for conducting a database table update, through the same set of API instances. In other words, according to the Burger patent, there is no need to develop and maintain different APIs for computer programs written in different languages.

There is a fundamental difference between the Burger patent and the present application. According to the Burger patent, a number of different individual computer application programs can each separately drive the operation of a system software by feeding appropriate instructions to the system software through the generic API. In the present application, on the other hand, a general-purpose software like the LearnApp drives the operation of individual computer application programs. If the instruction flow of the Burger patent and the present application were to be characterized using a client-server model, the flow of control is from the clients (computer application programs) to a server (the system software) in the Burger patent, but from a server (LearnApp) to clients (computer application programs) in the present application. In other words, Burger actually teaches away from what is recited in claim 1, as amended. Therefore, claim 1 and its dependent claims 2 and 5-7 are not anticipated by Burger.

Claim 11, as amended, is directed to a computer program product claim that includes limitations similar to those in claim 1. Therefore, claim 11 and its dependent claims 12 and 15-17 are not anticipated by Burger for at least the same reasons discussed above.

Claim rejections under 35 U.S.C. 103(a)

To reject claims in an application under 35 U.S.C. 103, the Examiner bears the initial burden of establishing a prima facie case of obviousness. *In re Bell*, 26 USPQ2d 1529, 1530 (Fed. Cir. 1993). In order to establish prima facie obviousness, the prior art, alone or in combination, must teach or suggest each and every limitation of the rejected claims. See *In*

re Vaeck, 20 USPQ2d 1438 (Fed. Cir. 1991); *In re Royka and Martin* 180 USPQ 580 (C.C.P.A. 1974); and *In re Wilson* 165 USPQ 494 (C.C.P.A. 1970). The teaching or suggestion to make the claimed invention, as well as the reasonable expectation of success, must come from the prior art, not Applicants' disclosure. *In re Vaeck, Id.*

In addition to Burger, the Examiner cited three additional references in the Office Action, contending that the combination of Burger and at least one of the three references renders at least one of the claims in the present application unpatentable.

However, for the reasons explained below, none of the three references cited by the Examiner teach or suggest a method of executing the first and second computer application programs using the first and second sets of control sequences stored in the data structure as recited in claim 1. Therefore, the Applicant respectfully traverses the rejections.

Dobbs teaches a method of determining, for a particular media type, which existing print mode of a printer's software or firmware driver will produce the highest print quality. In other words, given a print medium, e.g., a transparency, and an ink-jet print driver inherently having multiple print modes, Dobbs tries to identify a print mode most suitable for the print medium, i.e., the one that produces highest-quality prints. Neither Dobbs nor the combination of Burger and Dobbs teaches or suggests a process that extracts first and second control sequences from first and second programs, loading those sequences into a data structure, and then using the stored sequences to execute the first and second programs.

Ball is directed to a method of monitoring event queue operations. A customized event queue module processes a set of run-time messages resulting from an operation performed with a graphical user interface and reports operations associated with the set of messages. This can be used for debugging, tracing, and event recording. But there is no teaching or suggestion that the customized event queue module can execute multiple computer application programs using their respective sets of control sequences stored in a data structure. More generally, neither Ball nor the combination of Burger, Dobbs and Ball teaches or suggests a process that extracts first and second control sequences from first and second programs, loading those sequences into a data structure, and then using the stored sequences to execute the first and second programs.

Finally, Pham teaches a client-server message handling system which allows data types and data formats to be different at each client-end of the message handling system, and any changes in data elements, data types or data formats of the messages on the client-ends

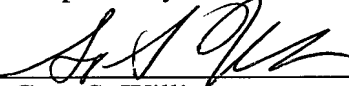
only require a reconfiguration of the server-end before start-up. Since reconfiguration is an administrative level activity, the user at a client-end will not be required to change his or her source code in the communicating applications. See Fig. 2 and col. 7, lines 49-65 of the Pham patent. Therefore, the instruction flow of the Pham patent is also from client to server which is similar to that of the Burger patent and opposite to that of the present application.

Since no combination of the cited references discloses every limitation of claim 1, claim 1 and its dependent claims 3-10 are patentable over the cited references. Claim 11, directed to a computer program product, and its dependent claims are patentable over the prior art of record for substantially the same reasons as claims 1-10.

In light of the above amendments and remarks, the Applicant respectfully requests that the Examiner reconsider this application with a view towards allowance. The Examiner is invited to call the undersigned attorney if a telephone call could help resolve any remaining items.

Respectfully submitted,

Date: September 30, 2004



Gary S. Williams

MORGAN, LEWIS & BOCKIUS LLP
2 Palo Alto Square, Suite 700
3000 El Camino Real
Palo Alto, California 94306
(650) 843-4000

31,066

(Reg. No.)